

A4

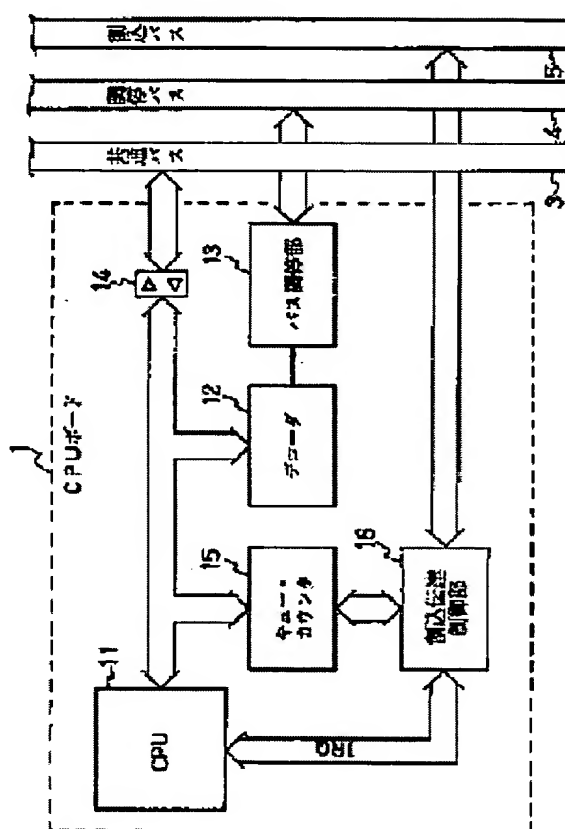
**INTERRUPTIVE PROCESSING DECENTRALIZATION SYSTEM**

**Patent number:** JP7244649  
**Publication date:** 1995-09-19  
**Inventor:** SAITO HIROYUKI  
**Applicant:** FUJITSU LTD  
**Classification:**  
 - international: G06F15/163; G06F9/46  
 - european:  
**Application number:** JP19940036660 19940308  
**Priority number(s):**

**Abstract of JP7244649**

**PURPOSE:** To decentralize the load on a device of loosely coupled type multiprocessor constitution by preventing interruptive processing from being concentrated on a specific processor.

**CONSTITUTION:** The device of loosely coupled type multiprocessor constitution obtained by connecting plural CPU boards 1, on which CPUs 11 are mounted, to a common bus 3 in parallel, is provided with queue counters 15 at interruptive processing parts on the respective CPU boards 1 to count queues of interrupts to the CPU boards 11; and interruption transmission control parts 16 are provided to control whether or not interrupts from the common bus are sent to the CPUs 11 according to the counting results of the interrupt queues, thereby transmitting the interrupts to the CPUs 11 only when the number of interrupts from the common bus is large.



Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-244649

(43) 公開日 平成7年(1995)9月19日

(51) IntCl.<sup>6</sup>

G 0 6 F 15/163  
9/46

識別記号

庁内整理番号

3 6 0 A 7737-5B

F I

技術表示箇所

G 0 6 F 15/ 16

3 1 0 Q

審査請求 未請求 請求項の数10 O L (全 12 頁)

(21) 出願番号 特願平6-36660

(22) 出願日 平成6年(1994)3月8日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 齋藤 宏行

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 柏谷 昭司 (外1名)

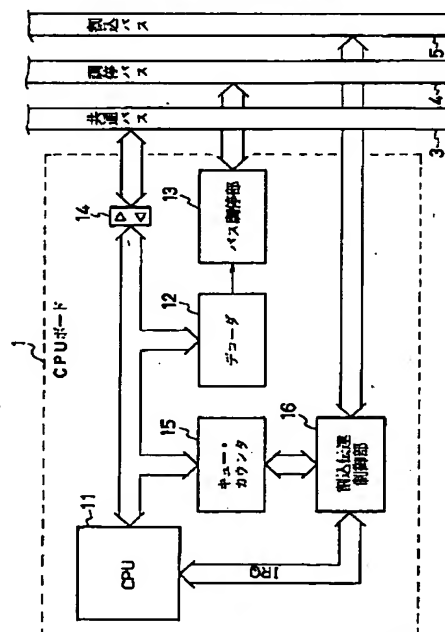
(54) 【発明の名称】 割込処理分散方式

(57) 【要約】

【目的】 疎結合型のマルチプロセッサ構成をとる装置において、特定のプロセッサに割込処理が集中することを防止して、負荷の分散を図ることを目的とする。

【構成】 CPU 11 を搭載した複数の CPU ボードを並列に共通バス 3 に接続してなる疎結合型のマルチプロセッサ構成の装置において、各 CPU ボードにおける割込処理部分に、キュー・カウンタ 15 を設けて、CPU に対する割込キューを計数し、割込伝達制御部 16 を設けて、割込キューの計数結果に応じて共通バスからの割込を CPU に伝達するか否かを制御することによって、共通バスからの割込数が多いときのみ CPU に割込を伝達するように構成する。

本発明の原理的構成 (1) を示す図



1

## 【特許請求の範囲】

【請求項 1】 CPU (11) を搭載した複数の CPU ボードを並列に接続してなる疎結合型のマルチプロセッサ構成の装置において、各 CPU ボードにおける割込処理部分に、

該 CPU に対する割込キューを計数するキュー・カウンタ (15) と、

該割込キューの計数結果に応じて CPU への割込の伝達を制御する割込伝達制御部 (16) とを設けて、共通バス (3) からの割込数が多いときのみ CPU に割込を伝達するようにしたことを特徴とする割込処理分散方式。

【請求項 2】 前記割込伝達制御部 (16) が、共通バス (3) からの割込を計数し、該割込の計数値と前記割込キューの計数値とを比較して、共通バスからの割込数が多いとき、割込を CPU に伝達することを特徴とする請求項 1 に記載の割込処理分散方式。

【請求項 3】 前記割込伝達制御部 (16) が、定常的にサイクリックに一定値をカウントするカウンタ (27) の計数値と前記割込キューの計数値とを比較して、該カウンタの計数値が多いとき、割込を CPU に伝達することを特徴とする請求項 1 に記載の割込処理分散方式。

【請求項 4】 前記割込伝達制御部 (16) が、定常的にサイクリックに一定値をカウントするカウンタ (27) の計数値と前記割込キューの計数値とをアドレスとして ROM (28) に格納されている値を読み出し、該読み出された値に応じて割込を CPU に伝達するウィンドウ時間を制御することを特徴とする請求項 1 に記載の割込処理分散方式。

【請求項 5】 前記割込伝達制御部 (16) が、定常的にサイクリックに一定値をカウントするカウンタ (27) の計数値と前記割込キューの計数値とをアドレスとして、装置の動作状況に対応して予めマッピングされている RAM (29) の値を読み出し、該読み出された値に応じて割込を CPU に伝達するウィンドウ時間を制御することを特徴とする請求項 1 に記載の割込処理分散方式。

【請求項 6】 CPU (11) を搭載した複数の CPU ボードを並列に接続してなる疎結合型のマルチプロセッサ構成の装置において、各 CPU ボードにおける割込処理部分に、

該 CPU に対する割込の制御と、ベクターのプリフェッチとを行う割込／ベクター・プリフェッチ制御部 (17) を設けて、

共通バス (3) からの割込入力時、バス調停を起動してバス権を得たとき割込元のベクターをラッチ (18) にプリフェッチし、CPU が割込を受け付けてスタック・フレームの退避等の処理を終了したとき該プリフェッチされたベクターを取り込んで割込処理を起動するように

2

したことを特徴とする割込処理分散方式。

【請求項 7】 前記割込／ベクター・プリフェッチ制御部 (17) が CPU (11) に内蔵されていることを特徴とする請求項 6 に記載の割込処理分散方式。

【請求項 8】 前記割込／ベクター・プリフェッチ制御部 (17) が CPU および割込処理部分に対する割込信号の中継をオン／オフするラッチ (31) を有し、該 CPU がホルト状態にないとき該ラッチ (31) を開いて割込信号の中継することを特徴とする請求項 6 または 7 に記載の割込処理分散方式。

【請求項 9】 前記割込／ベクター・プリフェッチ制御部 (17) が CPU および割込処理部分に対する割込信号の中継をオン／オフするラッチ (31) を有し、前記ラッチ (18) にベクターをラッチして CPU が割込処理を開始したとき該ラッチ (31) をオフにし、該割込処理を終了したとき該ラッチ (31) をオンにすることを特徴とする請求項 6 または 7 に記載の割込処理分散方式。

【請求項 10】 前記割込／ベクター・プリフェッチ制御部 (17) が CPU に対する割込信号の中継をオン／オフするバッファ (40) を有し、前記ラッチ (18) に有効なベクターがラッチされたとき該バッファ (40) をオンにして CPU に割込信号を入力し、該 CPU がスタック・フレームの退避等の処理を終了してベクター要求が発生したとき該ラッチ (18) 内のベクターを CPU に通知することを特徴とする請求項 6 または 7 に記載の割込処理分散方式。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、疎結合型のマルチプロセッサ構成をとる装置において、特定のプロセッサに割込処理が集中することを防止して負荷を均等化するための、割込処理分散方式に関するものである。

【0002】疎結合型のマルチプロセッサ構成をとる装置においては、共通バスを介して結合された複数のマルチプロセッサが、それぞれ独自に有するプログラムに基づいて独立に動作することができる。例えば疎結合型のマルチプロセッサ構成をとる無線通信用各種制御装置等においては、装置の状態変化や外部との通信等が非同期で発生するが、これに対する処理要求を並列に接続された各中央処理装置 (CPU) に通知するために、共通バスを介して、各 CPU に割込を通知することによって、割込に応答した CPU が所要の処理を行うようになっていく。

【0003】この場合、各 CPU が負荷を平等に分担することによって、過負荷がかかった場合でも、装置として最大の能力を発揮することができる。そのため、共通バスから入力される割込を、各 CPU ができるだけ均等に受け取るようにする必要がある。

【0004】このことは、システム構成の変化等によっ

3

て、装置の能力アップを図ろうとする場合には、並列に接続されるCPUの数を増加するだけで、簡易に対処可能にするためにも、必要なことである。

【0005】また上記の装置では、スレーブの入出力（I/O）盤から発生する割込を、各CPUに均等に処理させることによって、各CPUに加わる負荷を均等に分散させている。従って各CPUが割込に应答し、その割込処理を起動するためのベクターを、共通バス上のスレーブのI/Oポート等から均等に獲得するようにすることが必要である。

【0006】

【従来の技術】図11は、従来の装置構成例を示したものである。図中において、1<sub>1</sub>～1<sub>n</sub>はそれぞれ同等の構成を有するCPUボード、2はI/Oボード、3は共通バス、4は調停バス、5は割込バスである。各CPUボードにおいて、11はCPU、12はアドレスをデコードするデコーダ、13は共通バスのアクセス権を調停するバス調停部、14はバッファである。またI/Oボード2において、20はデータの入出力を制御するI/O制御部である。

【0007】図11において、各CPUボードのCPUには、割込バス5から割込（IRQ）が並列に入力されている。各CPUは割込を受け付けたとき、スタック・フレームの退避等を行って割込処理を起動すると、バス調停部13に共通バス獲得要求を出す。そしてバス調停の結果バス権を得たCPUは、割込元のI/Oボードから割込ベクター番号を受けとって、所定の割込要因処理タスクをキューとして積んで、割込処理からぬける。

【0008】割込要因処理タスクの順番は、OS（Operation System）によって管理実行され、順次、割込要因処理タスクが起動されて、受信電文の引取りや、状態変化の認識処理等が行われる。割込処理中は割込がマスクされて、他の割込が共通バスから入力されても、CPUは認識することができない。

【0009】またI/Oボード2からの割込は、共通バス3を介して各CPUボードに入力され、それぞれのCPUボードは、次のような処理フローに従って、ベクターをフェッチする。

【0010】

① 現在実行中の命令を終了する。...

↓

② プロセッサの内部状態をスタック・フレームに退避する。

↓

③ ベクター・フェッチのため、共通バス要求を起動する。

↓

④ ベクターをフェッチし、割込処理を起動する。ただしこの処理は共通バス調停の結果バス権を得たCPUのみであって、他のCPUは無効割込処理を起動する。

4

【0011】従って、各CPUに同じソフトウェアが搭載してあっても、その動作の同期がとれている訳ではないので、同時に割込が入力されても、各CPUがベクターをフェッチしにゆくまでの時間には、ばらつきがある。

【0012】

【発明が解決しようとする課題】CPUが割込信号を認識して、スタック退避等を行って割込処理を起動するのは、割込信号が入力されたときに実行中であった命令群を終了してからになるので、各CPUがプログラムのどの部分を走行していたかによって、実行中の命令が異なることとなり、割込信号が各CPUに同時に並列に入力されても、割込処理が起動されるまでの時間は、各CPUが実行中の命令によって異なる。

【0013】そこで、プログラムによっては、割込処理後に走行する部分の命令群のクロック数が少ないことがあって、割込に反応しやすくなるため、ある特定のCPUが割込を多くとるようになり、割込処理のキューを多数積んで負荷が集中する現象を生じて、負荷分散の観点から問題となっていた。

【0014】また、クロック数が少ない命令を実行している場合には、処理時間が比較的短くてすむため、ソフトウェアの構造によっては、ある一つのCPUのベクター・フェッチ動作が速くなる。これは、割込ハンドラーの部分で、少ないクロック数の命令で構成した場合等に生じる。そのため、そのCPUが割込処理を多数獲得してしまうことがあって、負荷分散の観点から問題となっていた。

【0015】本発明は、このような従来技術の課題を解決しようとするものであって、疎結合型のマルチプロセッサ構成をとる装置において、各CPUがプログラムのどの部分を走行していても、ある特定のCPUに割込が集中することを防止して、より負荷が軽いCPUに割込がかかりやすくし、従って負荷の均等分散が可能になるようにすることを目的としている。

【0016】また本発明は、CPUに搭載されているプログラムの構造のいかにかわらず、各CPUが均等にベクターをフェッチするようにして、割込処理が各CPUにおいて均等に起動され、従って、負荷の分散が可能になるようにすることを目的としている。

【0017】

【課題を解決するための手段】図1は、本発明の原理的構成（1）を示したものであって、図11における同じものを同じ番号で示している。CPUボード1において、15は割込要因処理タスクのキューをカウントするキュー・カウンタ、16はキュー・カウンタ値に応じて割込信号をCPU11に伝達する時間または数を制限制御する割込伝達制御部である。

【0018】キュー・カウンタ15は、割込要因処理タスクのキューの発生または消滅に応じて、CPU11に

5

よってカウントアップまたはカウントダウンするように制御される。割込伝達制御部 16 は、キュー・カウンタ 15 のカウント値に応じて、共通バス 3 から入力される割込信号を、CPU 11 に伝達する時間、または数を制限するように制御する。

【0019】図 1 において、キュー・カウンタ 15 は、CPU 11 が割込処理を起動し、割込要因処理タスクをキューに積んだときカウントアップし、割込要因処理タスクが起動し、処理が完了したとき、カウントダウンするように制御する。従って、キュー・カウンタ 15 に

は、現在積まれているキューの数がカウントされている。  
【0020】割込伝達制御部 16 は、キュー・カウンタ 15 のカウント値に応じて、共通バス 3 から入力される割込信号を、CPU 11 に伝達する数、または時間を制限するように制御する。すなわち、キュー・カウンタ 15 の値が大きくなるに従って、CPU 11 に通知する割込数を少なくしてキューを減少させる。または、キュー・カウンタ 15 の値が大きくなるに従って、CPU 11 に通知する割込のディレー時間を大きくして、割込処理が起動するのに要する時間を長くしてキューを減少させる。

【0021】図 2 は、本発明の原理的構成 (2) を示したものであって、図 11 における同じものを同じ番号で示している。CPU ボード 1 において、17 は割込の受付と、ベクター・フェッチの制御を行う割込/ベクター・プリフェッチ制御部、18 はベクターをプリフェッチするラッチ、19 は CPU のローカル・バスである。

【0022】割込/ベクター・プリフェッチ制御部 17 は、ローカル・バス 19 と共通バス 3 に接続されているとともに、共通バス 3 からの割込信号や、CPU 11 からの割込信号/割込制御信号等も接続されている。

【0023】割込/ベクター・プリフェッチ制御部 17 は、共通バス 3 からの割込信号の入力から一定時間後に共通バス要求を出力し、割込元の I/O ボード等からベクター番号をプリフェッチする。割込の入力に同期して、一定時間後に、各 CPU が一斉に共通バス要求を出力すると、各 CPU の割込/ベクター・プリフェッチ制御部 17 には、均等に割込ベクターがプリフェッチされる。各 CPU は、割込信号によって、割込ベクター要求を発生するので、割込/ベクター・プリフェッチ制御部 17 から、さきにプリフェッチしているベクターを CPU に渡すことによって、割込処理が起動される。

【0024】図 2 において、割込/ベクター・プリフェッチ制御部 17 には、まず共通バス 3 からの割込信号が入力される。CPU 11 は割込を受付け可能であるときは、割込信号が入力されてから、一定時間後に共通バス要求を出力し、バス調停を起動して共通バス 3 が獲得できた場合は、割込元の I/O ボード等からベクター番号

6

をプリフェッチする。共通バス 3 が獲得できなかった場合は、無効割込とする。

【0025】ここで、バス調停方式は、各 CPU に均等に共通バス獲得権が与えられているものとする。割込が入力されたのに同期して、一定時間後に各 CPU がベクター・プリフェッチのために、一斉に共通バス要求を出力することになるので、各 CPU の割込/ベクター・プリフェッチ制御部 17 には、均等に割込ベクターがプリフェッチされる。

【0026】また同時に、各 CPU の割込/ベクター・プリフェッチ制御部 17 は、割込信号を CPU 11 に中継しているので、CPU 11 は、実行中の命令が終了すると、内部状態をスタック・フレームに退避して割込ベクター要求を発生する。そこで、先に割込/ベクター・プリフェッチ制御部 17 にプリフェッチしているベクターを CPU 11 に渡すことによって、所定の割込処理を起動する。

【0027】従って、各 CPU には均等に割込処理が起動されることになるので、負荷の均等分散を実現することができる。

【0028】

【作用】

(1) 疎結合型のマルチプロセッサ構成の装置においては、CPU 11 を搭載した複数の CPU ボードが並列に接続されている。この場合、各 CPU ボードにおける割込処理部分に、キュー・カウンタ 15 を設けて、CPU に対する割込キューを計数し、割込伝達制御部 16 を設けて、割込キューの計数結果に応じて割込を CPU に伝達するか否かを制御するように構成して、共通バスからの割込数が多いときのみ CPU に割込を伝達するようにする。

【0029】本発明によれば、各 CPU がプログラムのどの部分を走行していても、蓄積されているキューの数に応じて、CPU に対する割込の伝達を、ハードウェアによって自律的にコントロールできるので、特定の CPU に割込が集中することがなく、より負荷の軽い CPU に割込がかかりやすくなるので、負荷の均等分散を図ることができる。

【0030】(2) この場合に、割込伝達制御部 16 が、共通バスからの割込を計数し、割込の計数値と割込キューの計数値とを比較して、共通バスからの割込数が多いとき、割込を CPU に伝達するようにしてもよい。

【0031】このようにすることによって、割込キュー数が増加するほど、CPU に割込が伝達される数が少なくなり、割込伝達が制限される。従って、割込はより負荷が小さい CPU にかかるようになり、負荷が分散される。

【0032】(3) この場合に、割込伝達制御部 16 が、定期的にサイクリックに一定値をカウントするカウンタ 27 の計数値と割込キューの計数値とを比較して、カウ

10

20

30

40

50

7

ンタの計数値が大きいいとき、割込をCPUに伝達するようにしてもよい。

【0033】このようにすることによって、より負荷が小さいCPUに割込がかかりやすくなり、負荷の分散を図ることができるようになる。

【0034】(4) この場合に、割込伝達制御部16が、定期的にサイクリックに一定値をカウントするカウンタ27の計数値と割込キューの計数値とをアドレスとしてROM28に格納されている値を読み出し、読み出された値に応じて割込をCPUに伝達するウィンドウ時間を制御するようにしてもよい。

【0035】このようにすることによって、割込キューが蓄積するほど、割込をCPUに伝達するウィンドウ時間が短くなるので、他の負荷が小さいCPUが割込をとる確率が高くなり、割込負荷の分散を図ることが可能となる。

【0036】(5) またこの場合に、割込伝達制御部16が、定期的にサイクリックに一定値をカウントするカウンタ27の計数値と割込キューの計数値とをアドレスとして、装置の動作状況に対応して予めマッピングされているRAM29の値を読み出し、読み出された値に応じて割込をCPUに伝達するウィンドウ時間を制御するようにしてもよい。

【0037】この場合、(4)の場合と同様に、割込負荷の分散を図ることが可能であるとともに、アプリケーションプログラムの性能等によって、ウィンドウ時間の設定を変えることができる。

【0038】(6) 疎結合型のマルチプロセッサ構成の装置においては、CPU11を搭載した複数のCPUボードが並列に接続されている。この場合、各CPUボードにおける割込処理部分に、割込／ベクター・プリフェッチ制御部17を設けて、CPUに対する割込の制御と、ベクターのプリフェッチとを行うように構成して、共通バス3からの割込入力時、バス調停を起動してバス権を得たとき割込元のベクターをラッチ18にプリフェッチし、CPUが割込を受け付けてスタック・フレームの退避等の処理を終了したとき、プリフェッチされたベクターを取り込んで割込処理を起動するようにする。

【0039】本発明によれば、CPUに搭載されているプログラムの構造のいかんにかかわらず、各CPUが均等にベクターをフェッチするようにしたので、割込処理が各CPUにおいて均等に起動され、従って、負荷の完全分散が可能になる。

【0040】(7) この場合、割込／ベクター・プリフェッチ制御部17を、CPU11に内蔵するようにしてもよい。

【0041】(8) これらの場合、割込／ベクター・プリフェッチ制御部17が、CPUおよび割込処理部分に対する割込信号の中継をオン／オフするラッチ31を有し、CPUがホルト状態にないとき、ラッチ31を開い

8

て割込信号の中継するようにしてもよい。

【0042】このように、割込発生時、ハードウェア処理によって割込をラッチして、共通バス獲得の調停を行うので、バス調停が各プロセッサに均等にバス獲得権が与えられる方式の場合、割り込み発生に同期して、各CPUが同時にバス要求を出すので、各CPUには均等にベクターがプリフェッチされ、各CPUは均等に割込処理を起動することができ、負荷の均等分散を実現することができる。

【0043】(9) これらの場合、割込／ベクター・プリフェッチ制御部17が、CPUおよび割込処理部分に対する割込信号の中継をオン／オフするラッチ31を有し、ラッチ18にベクターをラッチしてCPUが割込処理を開始したとき、ラッチ31をオフにし、割込処理を終了したとき、ラッチ31をオンにするようにしてもよい。

【0044】このようにすることによって、割込が多発した場合には、割込イベントが発生するごとに、バス調停に参加するCPUが減少するので、より厳密に負荷の均等分散を図ることができる。

【0045】(10) またこれらの場合、割込／ベクター・プリフェッチ制御部17がCPUに対する割込信号の中継をオン／オフするバッファ40を有し、ラッチ18に有効なベクターがラッチされたとき、バッファ40をオンにしてCPUに割込信号を入力し、CPUがスタック・フレームの退避等の処理を終了してベクター要求が発生したとき、ラッチ18内のベクターをCPUに通知するようにしてもよい。

【0046】このようにすることによって、CPUに割込信号が通知されて、CPUが内部状態をスタック・フレームに退避してベクター要求を出したときは、必ず有効なベクターをCPUに渡すことができるので、CPUが無効割込処理を行うことがなくなり、CPU動作の性能向上を図ることが可能となる。

【0047】

【実施例】図3は、本発明の実施例(1)を示したものであって、CPU11に対する割込処理部分を示し、図1における同じものを同じ番号で示している。21はカウント・コントロール・レジスタであって、キューの発生または消滅に応じて、アップクロックまたはダウンクロックを出力する。22はアップ／ダウン・カウンタであって、積まれているキューの数をカウントする。23は割込数カウンタであって、共通バスにおける割込入力数をカウントする。24は、コンパレータであって、アップ／ダウン・カウンタ22のカウント値と、割込数カウンタ23のカウント値とを比較する。25はゲートであって、コンパレータ24の出力に応じて、イネーブルとなりまたはデセーブルとなる。26はオア回路であって、共通バスからの各割込信号の論理和を求める。

【0048】CPU11は、カウント・コントロール・

9

レジスタ 21 を介して、割込処理が起動し割込要因処理タスクをキューに積んだとき、アップ/ダウン・カウンタ 22 をアップカウントする。また割込要因処理タスクが起動し処理を完了したとき、ダウンカウントする。従ってアップ/ダウン・カウンタ 22 には、積まれているキューの数がカウントされている。

【0049】一方、割込数カウンタ 23 は、オア回路 26 を経て共通バスから割込が入力されるごとに、カウンタアップされる。コンパレータ 24 は、アップ/ダウン・カウンタ 22 の値 (n) と、割込数カウンタ 23 の値 (m) とを比較して、 $m > n$  の場合、ゲート 25 をイネーブルにして、共通バスからの各割込を CPU 11 へ通知する。なお、割込数カウンタ 23 は、CPU 11 が割込処理を起動すると、割込応答 (IACK) 信号によってリセットされる。

【0050】従って、キューが蓄積して n 個になったとき、CPU 11 に割込が伝達される数は、 $1/(n+1)$  となり、割込伝達が制限される。これによって、より負荷のかかっていない CPU に割込がかけられるようになり、負荷が分散される。

【0051】図 4 は、本発明の実施例 (2) を示したものであって、図 3 における同じものを同じ番号で示し、27 は定常的にサイクリックに一定数をカウントアップまたはカウントダウンするカウンタである。

【0052】アップ/ダウン・カウンタ 22 においては、図 3 の場合と同様にして、現在蓄積されているキューの数がカウントされる。カウンタ 27 は、定常的に一定数までカウントアップし、または一定数からカウントダウンを繰り返している。コンパレータ 24 は、アップ/ダウン・カウンタ 22 の値 (n) と、カウンタ 27 の値 (m) とを比較して、 $m > n$  のとき、ゲート 25 をイネーブルにして、共通バスからの割込信号を CPU 11 に伝達する。従って、キューの蓄積数が増加すると、CPU 11 に割込が通知されるウィンドウ時間が短くなり、CPU 11 に入力される割込の数が減少する。

【0053】このように、図 4 に示された実施例では、より負荷の小さい CPU に割込がかけられやすくなるので、負荷の分散を図ることが可能となる。

【0054】図 5 は、本発明の実施例 (3) を示したものであって、(a) は回路構成、(b) はリードオンリーメモリ (ROM) の構成を示している。また図 6 は、ROM の内容の例を示したものである。図 4 における同じものを同じ番号で示し、28 は ROM であって、アップ/ダウン・カウンタ 22 の値と、カウンタ 27 の値とをアドレスとして出力を発生する。

【0055】アップ/ダウン・カウンタ 22 においては、図 3 の場合と同様にして、現在蓄積されているキューの数がカウントされる。カウンタ 27 は、定常的にカウントアップまたはカウントダウンを繰り返している。ROM 28 は、図 6 に示すように、カウンタ 27 のカウ

10

ント値を下位アドレス ( $A0 \sim An$ ) に入力され、アップ/ダウン・カウンタ 22 のカウント値を上位アドレス ( $An+1 \sim Am$ ) に入力されることによって、単位時間の  $1/16$  ごとに順次 00 または 01 となる出力  $D_0$  を発生する。

【0056】例えば、カウンタ 27 のカウント値が 00 であって、アップ/ダウン・カウンタ 22 のカウント値が 00 のときは、ROM 28 の出力  $D_0$  は、すべての時間 01 であって、ゲート 25 は常に開いている。アップ/ダウン・カウンタ 22 のカウント値が 10 のときは、ROM 28 の出力  $D_0$  は、最初の  $1/16$  の時間 00 であってゲート 25 が閉じ、以後 01 となってゲート 25 が開く。アップ/ダウン・カウンタ 22 のカウント値が 20 のときは、ROM 28 の出力  $D_0$  は、始めの 2 つの  $1/16$  の時間 00 であってゲート 25 が閉じ、以後 01 となってゲート 25 が開く。カウンタ 27 の値が 00 以外の値のときも同様に、ゲート 25 の開閉の関係を定めることができる。

【0057】このように、ROM 28 の内部に、図 6 に示すようなデータを格納しておくことによって、アップ/ダウン・カウンタ 22 の値が増加 (減少) するのに伴って、ゲート 25 をイネーブルにする期間を減少 (増加) させることができる。

【0058】従って図 5 の実施例によれば、割込キューが蓄積するのに伴って、割込信号を CPU 11 に対して伝達するウィンドウ時間が短くなり、他の負荷がかかっていない CPU が割込をとる確率が高くなるので、割込負荷の分散を図ることができる。

【0059】図 7 は、本発明の実施例 (4) を示したものであって、(a) は回路構成、(b) はランダムアクセスメモリ (RAM) の構成を示している。図 5 における同じものを同じ番号で示し、29 は RAM であって、アップ/ダウン・カウンタ 22 の値と、カウンタ 27 の値とをアドレスとして出力を発生する。RAM 29 には、図 6 に示された ROM 28 と同様の内容を予めマッピングされている。30 は双方向ゲートであって、立ち上げ時に RAM 29 の内容を書き込むために、共通バスを RAM 29 に接続する。

【0060】アップ/ダウン・カウンタ 22 においては、図 3 の場合と同様にして、現在蓄積されているキューの数がカウントされる。カウンタ 27 は、定常的にサイクリックに一定数のカウントアップまたはカウントダウンを繰り返している。RAM 29 は、カウンタ 27 のカウント値を下位アドレス ( $A0 \sim An$ ) に入力され、アップ/ダウン・カウンタ 22 のカウント値を上位アドレス ( $An+1 \sim Am$ ) に入力されることによって、単位時間の  $1/16$  ごとに順次 00 または 01 となる出力  $D_0$  を発生する。

【0061】CPU ボードのイニシャル処理時等において、アプリケーションプログラムの性能等の要件によつ

11

て、RAM 29の内部を予めマッピングしておくことによって、図5の場合と同様に、アップ/ダウン・カウンタ22の値が増加(減少)するのに伴って、ゲート25をイネーブルにする期間を減少(増加)させることができる。

【0062】このように図7の実施例によれば、割込キューが蓄積するのに伴って、割込信号をCPU11に対して伝達するウィンドウ時間が短くなり、従って、他の負荷がかかっていないCPUが割込をとる確率が高くなるので、割込負荷の分散を図ることができる。

【0063】図8は、本発明の実施例(5)を示したものであって、CPU11に対する割込処理部分を示し、図2における同じものを同じ番号で示している。図中、31はラッチ、32は割込信号に応じて一定時間後に各種制御信号を出力するタイマ回路、33は共通バス獲得の調停を行うバス調停部、34、35は双方向バッファ、36はローカルバスである。

【0064】CPU11が停止(halt)状態でないとき、すなわち割込処理を正常に行える場合は、ラッチ31を開いて、割込信号をタイマ回路32とCPU1120に入力する。タイマ回路32は、割込信号が入力されてから、一定時間後、共通バス獲得のため、バスリクエスト(BR)信号をバス調停部33へ送り、バス調停部33は、共通バス獲得の調停を行う。

【0065】共通バスが獲得されたとき、バス調停部33からのバス獲得(WIN)信号に応じて、タイマ回路32は割込制御信号を双方向バッファ34を経て共通バス3へ送出し、これによって割込元のI/Oボード等から、共通バス3を経て送出された割込ベクターを、双方向バッファ35、ローカルバス36を介してラッチ1830に取り込む。このときは、ラッチ31において割込信号もラッチし、CPU11が割込動作を起動して割込ベクター要求を発生した時点で、ラッチを解くものとする。

【0066】そして、CPU11がスタック・フレームへの内部状態の退避等の処理を完了して、ベクターをフェッチできるようになると、割込応答(IACK)信号を送出することによって、さきほどラッチ18に取り込まれたベクターが、ローカルバス36を経てCPU11に通知され、これによって、CPU11における所定の割込処理が起動される。

【0067】なお、共通バス調停時にバス権を獲得できなかったときは、無効割込として、ラッチ18にはベクターがプリフェッチされず、CPU11が割込ベクターを要求したときは、タイマ回路32からバスエラー(BERR)信号等を返すので、CPU11において、無効割込処理が起動される。

【0068】図9は、本発明の実施例(6)を示したものであって、図8における同じものを同じ番号で示し、38はフリップフロップ(FF)、39はバッファである。

12

【0069】割込ベクターのプリフェッチ、CPU11へのベクターの通知方法等は、図8の場合と同様にして行われる。図9の実施例においては、共通バスからの割込信号を、タイマ回路32とCPU11に中継するラッチ31をオン、オフする制御を、フリップフロップ38を介して行い、このフリップフロップ38のセット/リセットを、バッファ39を介してソフトウェアによって行えるようにしている。

【0070】これによって、割込ベクターを受けて割込処理を開始した時点で、ラッチ31をオフにし、割込処理を終了した時点でラッチ31をオンにするように制御すれば、割込が多発した場合には、割込イベントが発生するたびに、調停に参加するCPUの数が減少するので、より厳密に負荷の均等分散を図ることができるようになる。

【0071】図10は、本発明の実施例(7)を示したものであって、図8における同じものを同じ番号で示し、40は割込信号を中継するバッファである。

【0072】割込ベクターのプリフェッチ、CPU11へのベクターの通知方法等は、図8の場合と同様にして行われる。図10の実施例においては、CPU11へ割込信号の中継を行うバッファ40の制御を、ラッチ18に有効なベクターがラッチされたときオンにし、その他のときはオフにするようにしておく。

【0073】これによって、CPU11に割込信号が通知されて、CPU11が内部状態をスタック・フレームに退避し、ベクター要求を発生したときには、必ず有効なベクターをCPU11に渡すことができるので、CPU11が無効割込処理を行う必要がなくなり、CPU動作の性能向上を図ることができるようになる。

【0074】

【発明の効果】以上説明したように本発明によれば、疎結合型マルチプロセッサ構成をとる装置の各CPUにおいて、CPUがプログラムのどの部分を走行していても、共通バスからの割込信号がCPUに伝達されるのを、蓄積されているキューの数に応じて、ハードウェアによって自律的にコントロールしているので、ある特定のCPUに割込が多数取り込まれることはなく、より負荷が軽いCPUに割込がかかりやすくなっているため、負荷の均等分散を図ることが可能となり、従って負荷分散制御のオーバーヘッドが少ないとともに、この場合のハードウェアの実現規模も小さくてすむ。

【0075】また本発明によれば、CPUに搭載されているソフトウェアの構造のいかんによらず、各CPUは均等にベクターをフェッチすることができる。従って、割込処理はそれぞれのCPUで均等に起動され、負荷の完全分散を行うことが可能となる。

【図面の簡単な説明】

【図1】本発明の原理的構成(1)を示す図である。

【図2】本発明の原理的構成(2)を示す図である。

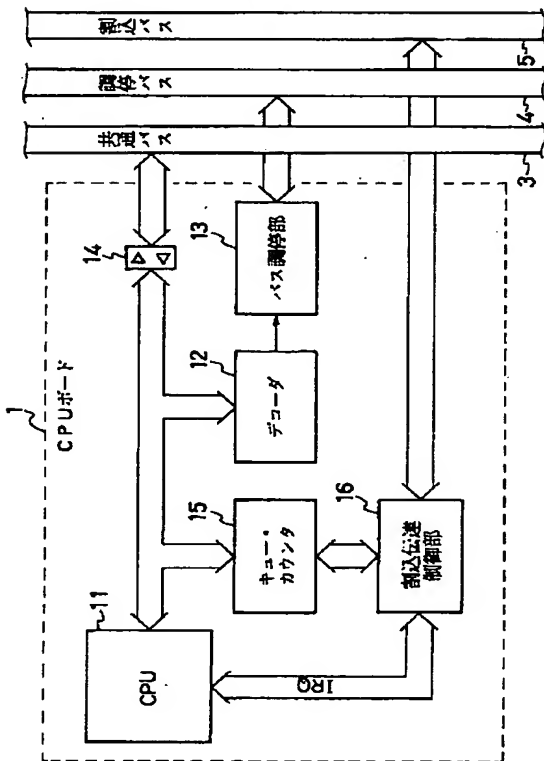


13

- 【図 3】 本発明の実施例 (1) を示す図である。  
 【図 4】 本発明の実施例 (2) を示す図である。  
 【図 5】 本発明の実施例 (3) を示す図であって、  
 (a) は回路構成、(b) はリードオンリーメモリ (ROM) の構成を示す。  
 【図 6】 ROM の内容の例を示す図である。  
 【図 7】 本発明の実施例 (4) を示す図であって、  
 (a) は回路構成、(b) はランダムアクセスメモリ (RAM) の構成を示す。  
 【図 8】 本発明の実施例 (5) を示す図である。  
 【図 9】 本発明の実施例 (6) を示す図である。  
 【図 10】 本発明の実施例 (7) を示す図である。  
 【図 11】 従来の装置構成例を示す図である。

【図 1】

本発明の原理的構成 (1) を示す図



14

## \* 【符号の説明】

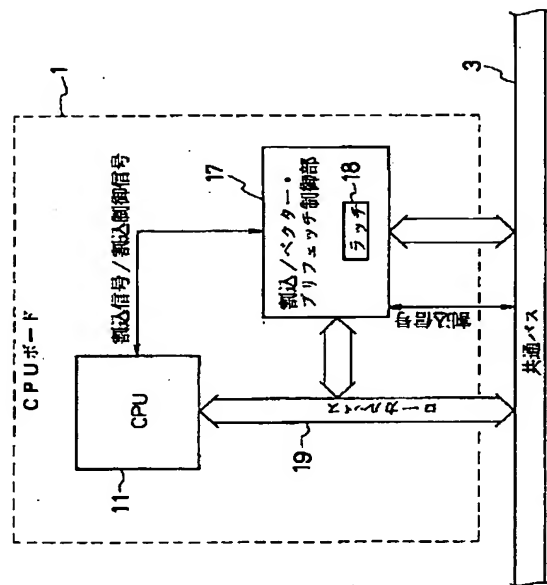
- 3 共通バス  
 11 CPU  
 15 キュー・カウンタ  
 16 割込伝達制御部  
 17 割込/ベクター・プリフェッチ制御部  
 18 ラッチ  
 27 カウンタ  
 28 ROM  
 29 RAM  
 31 ラッチ  
 40 パッファ

10

\*

【図 2】

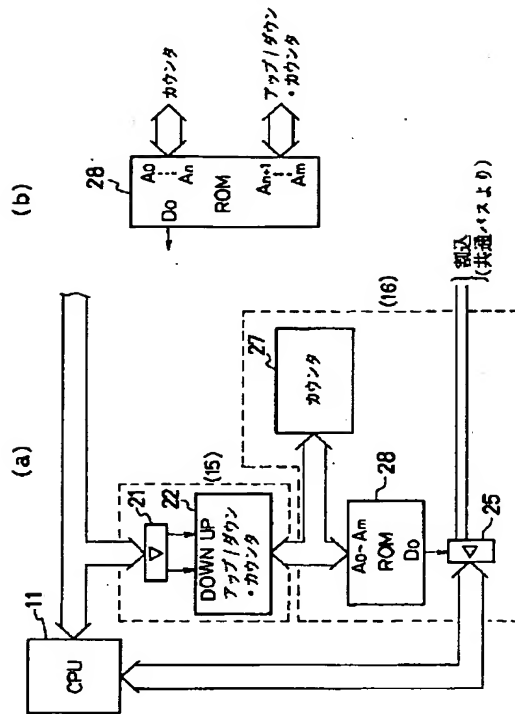
本発明の原理的構成 (2) を示す図





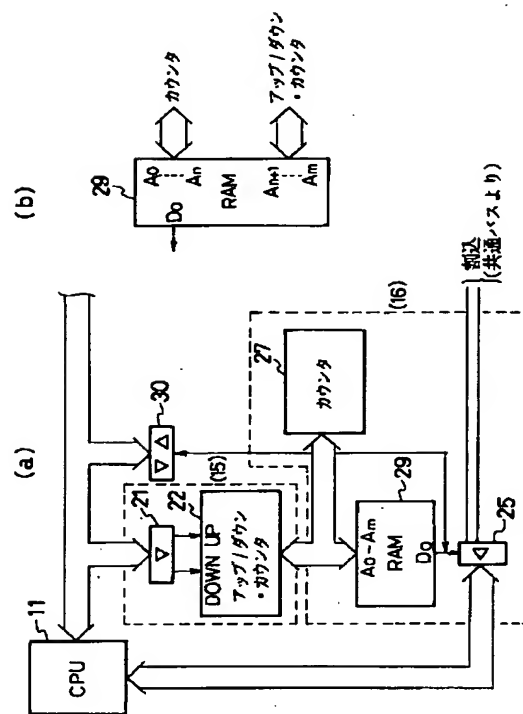
【図 5】

本発明の実施例（３）を示す図



【図 7】

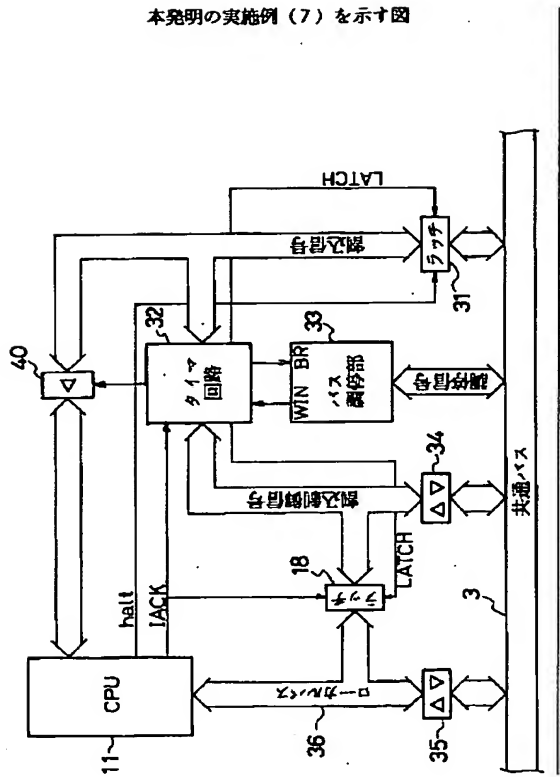
本発明の実施例（４）を示す図





【図 10】

本発明の実施例（７）を示す図



【図 11】

従来の装置構成例を示す図

